

ANDRe, una aplicació lliure per a combinar pàgines de documents en format PDF

*Projecte final del Màster de Programari Lliure
Desenvolupament d'aplicacions de programari lliure*

©2011 Domènec Martín Martínez
Aquesta obra està llicenciada sota una llicència Creative Commons
Reconeixement-CompartirIgual 3.0 Espanya¹

¹<http://creativecommons.org/licenses/by-sa/3.0/es/legalcode.ca>

Resum

Aquest projecte consisteix en el desenvolupament d'ANDRe, una aplicació de programari lliure que permet crear, de forma visual, documents PDF combinant pàgines d'altres documents del mateix format. L'aplicació ha estat desenvolupada en Java i utilitza la biblioteca *PDFBox* [1] de l'Apache Foundation per a llegir i escriure els documents.

El projecte inclou, a part de l'aplicació de programari en si, l'alliberament d'aquesta i la preparació d'un entorn de col·laboració adequat amb la finalitat d'atraure la comunitat, tant d'usuaris com de possibles col·laboradors. Aquests dos objectius s'han assolit utilitzant els serveis del lloc web de desenvolupament SourceForge[2], on s'allotja ANDRe des del seu inici[3].

Aquesta memòria comença exposant els objectius plantejats per al projecte, incloent un petit anàlisi d'una selecció d'aplicacions semblants a ANDRe, continua amb un estudi d'algunes eines de gestió de projectes de programari, resumeix la fase de desenvolupament i acaba exposant-ne els resultats i les conclusions.

En finalitzar aquesta memòria, ANDRe, versió *0.1 Alpha*, està alliberat sota la llicència GNU GPLv3[4]. Per a poder funcionar, necessita una màquina virtual de java compatible amb la versió 6 de Java (*Java SE 6*).

Índex

1	Introducció	7
1.1	Objectius	7
1.1.1	Estructura dels objectius específics	8
1.2	Abast i limitacions	9
1.2.1	Formats	9
1.2.2	Altres limitacions	9
1.3	Projectes existents similars	10
1.3.1	PDF-Shuffler	10
1.3.2	PDF Split and Merge (PDFsam) Basic Version	10
1.3.3	PDF Split and Merge (PDFsam) Enhanced Version .	11
1.3.4	PdfMod	11
1.3.5	PdfJumbler	11
1.3.6	Taula comparativa	12
1.4	Planificació temporal	12
1.4.1	Fites temporals	14
1.4.2	Diagrama de Gantt	14
1.5	Estructura de la memòria	15
2	Eines de gestió de projectes	17
2.1	Control de versions	18
2.1.1	CVS	18
2.2	Sistemes de seguiment d'errors	19
2.2.1	Bugzilla	19
2.3	Llistes de correu electrònic	20
2.3.1	Mailman	21
2.4	La gestió d'un projecte	21
2.4.1	Llicència	22
2.4.2	Atraure la comunitat	22
3	Realització del projecte	23
3.1	Programari, llenguatges i serveis utilitzats	23
3.1.1	Java	23
3.1.2	Eclipse	23

3.1.3	SourceForge	24
3.1.4	L ^A T _E X	24
3.2	El·lecció de les biblioteques	24
3.3	El·lecció de la llicència	25
3.4	Fases del desenvolupament del programa	26
3.4.1	Capa abstracta de lectura i creació de documents	26
3.4.2	Lectura i creació de documents PDF	28
3.4.3	Interfície gràfica d'usuari	28
3.5	Proves	29
3.5.1	Errors detectats	30
3.5.2	Mancances de funcionalitats bàsiques	30
3.6	Localització	31
3.6.1	Escriure una traducció	31
3.7	Publicació i promoció	32
4	Resultats	35
4.1	Finestra principal	35
4.2	Obrir documents	35
4.3	Crear un nou document	37
4.4	Sortir del programa	40
5	Conclusions	43
5.1	Assoliment d'objectius	43
5.2	ANDRe en el futur	45
5.3	Nota personal	45
	Referències	47
A	Recull de classes, interfícies i enumeracions	49

Capítol 1

Introducció

En el dia a dia d'una oficina es treballa sovint amb documents que, ja sigui per haver estat creats per terceres persones o pel seu format, són difícils d'editar. L'exemple més quotidià d'aquet fet és, potser, el treball amb documents PDF. Aquests són documents pensats per a ser únicament llegits, no editats, però a vegades sorgeix la necessitat de fer-ne modificacions a nivell de pàgina (eliminació o reordenació de pàgines, composició d'un document a partir de les pàgines d'altres documents, etc.). Alguns casos d'ús són:

- Es vol enviar una selecció de pàgines d'una llista de preus a un client.
- Es vol fusionar en un sol document, documents que estan estretament relacionats entre sí, però eliminant-ne la pàgina de títol de cadascun.
- Es volen reordenar les pàgines d'un document escanejat en un ordre incorrecte.
- Cal afegir o eliminar d'un document escanejat un pàgina oblidada o escanejada dues vegades, respectivament.

Amb aquest projecte es pretén realitzar un programa alternatiu als ja existents que permeti combinar i reordenar pàgines de diversos documents PDF i que doni el resultat en un nou document PDF. El programa s'anomenarà *ANDRe*, que vol dir *ANDRe's Not a Document Rearranger*.

1.1 Objectius

Amb la realització d'aquest projecte es pretén desenvolupar una aplicació informàtica senzilla, lliure i multiplataforma que ofereixi una solució als problemes plantejats en l'apartat anterior. Aquesta aplicació ha de permetre a l'usuari crear un document a partir de pàgines de documents ja existents, d'una manera visual i intuïtiva. Això vol dir que l'usuari podrà visualitzar imatges en miniatura (*thumbnails*) de les pàgines dels documents d'origen i

podrà construir el nou document arrossegant amb el ratolí aquests *thumbnails* tot col·locant-los en l'ordre desitjat. A més, durant aquest procés, l'usuari podrà veure en temps real com resultarà el nou document, per a la qual cosa els *thumbnails* s'hauran de desplaçar suaument mostrant contínuament el resultat. L'aplicació podrà oferir característiques complementàries, com ara poder desfer o refer els canvis efectuats al document, però es prioritzarà sempre que el programa sigui fàcil d'utilitzar i estigui lliure d'errors.

El fet que l'aplicació sigui lliure, però, afegeix uns altres objectius a més del de desenvolupar l'aplicació en si, ja que cal preparar una estratègia d'alliberament de l'aplicació que atregui la comunitat de desenvolupadors.

A més, per a facilitar l'ús del programa per part dels usuaris, s'hauran d'oferir unes instruccions precises d'obtenció i instal·lació d'aquest així com una documentació d'ús.

També amb la intenció d'arribar al major nombre de desenvolupadors i usuaris possible, s'utilitzarà l'anglès en les comunicacions i documentació durant el desenvolupament del projecte i es prepararà el codi de manera que sigui fàcilment internacionalitzable.

Finalment, i perquè ANDRe sigui multiplataforma, el desenvolupament es realitzarà en el llenguatge de programació Java i s'evitarà utilitzar biblioteques de plataformes específiques. L'elecció de Java, a més, fa que el nombre de potencials col·laboradors sigui molt gran, ja que és un dels llenguatges de programació més populars[5][6].

1.1.1 Estructura dels objectius específics

Objectius referents a la interfície d'usuari de l'aplicació

- Fer una interfície senzilla (fàcil d'utilitzar).
- La interfície ha de ser intuïtiva i atractiva (desplaçament suau dels *thumbnails*).
- Oferir característiques complementàries (desfer i refer).

Objectius tècnics

- Escriure el codi en Java.
- Evitar utilitzar biblioteques de plataformes específiques.
- Mantenir un nivell d'abstracció que permeti, en el futur, afegir compatibilitat amb altres formats (veure 1.2.1).
- Triar una llicència lliure adequada tenint en compte les biblioteques que es volen utilitzar.

Objectius d'acostament a la comunitat

- Oferir un sistema de control de versions amb possibilitat de lectura anònima.
- Oferir una llista de correu destinada als desenvolupadors.
- Oferir un fòrum per als desenvolupadors i un per als usuaris.
- Oferir un sistema de seguiment d'errors.
- Tenir un lloc web per al projecte, ben estructurat i amb elements atractius (captures de pantalla).
- Escriure el codi de forma ordenada i documentar-lo bé.
- Explicar detalladament com obtenir el codi i quins passos s'han de seguir per a fer-lo funcionar.
- Utilitzar l'anglès en les comunicacions i la documentació.
- Fer el programa fàcilment internacionalitzable
- Escriure una documentació d'ús.

1.2 Abast i limitacions**1.2.1 Formats**

En un principi, l'aplicació podrà treballar únicament amb documents de tipus PDF, però com que és desitjable que es puguin combinar documents amb diferents formats (per exemple, PostScript o DVI) i, fins i tot, imatges, es procurarà mantenir un nivell d'abstracció que permeti la inclusió futura d'altres formats. En qualsevol cas, la compatibilitat amb formats diferents del PDF queda fora de l'abast d'aquest projecte.

ANDRe no funcionarà, tampoc, amb tots els fitxers PDF: aquest és un format en constant evolució i és d'esperar que les biblioteques que s'empraran no seran compatibles amb les últimes versions. A més, hi ha PDFs amb característiques especials, com ara PDFs encriptats, la manipulació dels quals no està prevista entre les capacitats del programa.

1.2.2 Altres limitacions

- ANDRe està pensat per a realitzar petites i ràpides edicions i combinacions de documents. Per això, no serà possible desar l'estat d'un treball per a continuar-lo més tard.

- En la primera versió del programa, la optimització d'aquest no serà una prioritat. Això vol dir que el programa podria ser *massa* lent amb documents grans.

1.3 Projectes existents similars

A continuació es presenten algunes aplicacions lliures que permeten l'edició de documents PDF amb operacions a nivell de pàgina. Després es mostraran, en una taula comparativa, les característiques més importants exposades en el resum de cadascuna de les aplicacions, possiblement complementades amb informació addicional.

1.3.1 PDF-Shuffler[7]

Aquesta és una aplicació molt similar a la que es pretén desenvolupar en aquest projecte. És una aplicació amb una interfície molt simple que mostra *thumbnails* de les pàgines dels documents que haguem carregat. Permet reordenar les pàgines arrossegant-les amb el ratolí però sense mostrar el resultat en temps real. També permet eliminar, girar o retallar les pàgines seleccionades (accepta selecció múltiple), però no té l'opció de desfer els canvis efectuats. Els documents es poden carregar mitjançant un menú de fitxer o arrossegant-los amb el ratolí dins la finestra del programa.

PDF-Shuffler està programat en Python, cosa que el fa extremadament lent quan edita fitxers grans, i, segons la informació de SourceForge, funciona en tots els sistemes operatius POSIX. S'ofereix en forma de codi font amb un script per a instal·lar-lo i també en un paquet tipus *deb*. Té la llicència GNU GPLv2 i forma part del rebost d'Ubuntu 10.04.

1.3.2 PDF Split and Merge (PDFsam) Basic Version[8]

Es tracta d'una aplicació molt flexible per a realitzar diversos tipus d'edicions en documents PDF. Està basada en *plugins*, de forma que cada un aporta una funcionalitat diferent. En aquesta versió, la bàsica, només hi ha dos *plugins*: un serveix per a ajuntar diversos PDFs en un, i l'altre serveix per trencar un PDF en diversos documents seguint un criteri que es pot escollir (separar totes les pàgines, tallar el document per una pàgina donada, etc.). En l'opció d'ajuntar diversos documents, es pot triar l'ordre entre ells, però no l'ordre intern de les pàgines. Cap dels dos *plugins* ofereix un entorn visual (no hi ha *thumbnails*) i no es poden importar documents arrossegant-los amb el ratolí.

Aquesta aplicació està desenvolupada en Java i, en contrast amb PDF-Shuffler, genera els documents de sortida molt ràpidament. En la pàgina web oficial del projecte, s'ofereixen versions per a Windows i Mac OS X juntament amb el codi font, però cap paquet per a distribucions de Linux.

La distribució Ubuntu 10.04, però, l'ofereix en el seu rebost. Té la llicència GNU GPLv2.

1.3.3 PDF Split and Merge (PDFsam) Enhanced Version[8]

La versió avançada de PDFsam, anomenada Enhanced Version, és com la bàsica però incloent molts més *plugins*. Dos d'aquests *plugins*, combinats, aporten la mateixa funcionalitat que PDF-Shuffler. Un d'ells, el *Visual document composer*, permet importar diversos PDFs i seleccionar-ne pàgines de forma individual per a incloure-les en el document de sortida. Aquest procés, però, no es fa arrossegant les pàgines amb el ratolí, sinó que s'han de seleccionar primer i aleshores clicar sobre un botó. La reordenació de les pàgines en el document resultant es fa de manera similar. El *plugin Visual reorder* permet reordenar les pàgines exactament com l'aplicació PDF-Shuffler, però només d'un document alhora. En qualsevol cas, no hi ha cap possibilitat de desfer els canvis efectuats.

A diferència de la versió simple, en la pàgina oficial no s'ofereix cap versió binària. Només està disponible el codi font, sense instruccions per a compilar-lo ni executar el programa. La distribució Ubuntu 10.04 no l'ofereix en el rebost de programari, però a la pàgina http://wiki.ubuntuusers.de/PDF_Split_and_Merge hi ha instruccions per a instal·lar-lo (en alemany). Utilitza la llicència GNU GPLv2.

1.3.4 PdfMod[9]

Aquest és un programa molt similar al PDF-Shuffle, però amb una interfície d'usuari més treballada. A diferència de l'anterior, importa cada document en una finestra nova, permet desfer els canvis efectuats i permet modificar la mida dels *thumbnails*.

PdfMod està programat en C# (Mono) i, com tots els altres, té la llicència GNU GPLv2. Està disponible en paquets per a diverses distribucions de Linux, entre les quals Ubuntu, OpenSUSE i Fedora. En Ubuntu, no està en el rebost de la versió 10.04 però sí en el de la 10.10.

1.3.5 PdfJumbler[10]

És un programa molt senzill escrit en Java amb unes característiques similars a PdfMod, però amb un gran inconvenient en comparació amb tots els programes tractats fins ara: mostra els *thumbnails* en una columna en lloc d'una graella, provocant que fins i tot els documents curts no puguin ser mostrats completament a la pantalla.

Està publicat sota llicència AGPL. No es publica empaquetat per a cap distribució de Linux i no forma part del rebost de programari d'Ubuntu.

1.3.6 Taula comparativa

En la següent taula es comparen les principals característiques de PDF-Shuffler, PDFsam Enhanced Version, PdfMod i PdfJumbler. A més, s'afegeix una columna amb algunes característiques previstes de l'aplicació ANDRe.

	PDF-Shuffler	PDFsam	PdfMod	PdfJumbler	ANDRe
<i>Thumbnails</i>	Si	Si	Si	Si	Si
<i>Thumbnails</i> en graella	Si	Si	Si	No	Si
Prev. temps real	No	No	No	No	Si
Zoom	No	Si	Si	Si	? ¹
Desfer i refer	No	No	Si	Si	Si
Obrir arrossegant	Si	No	Si	Si	? ²
Edició pàgina	Girar, eliminar	Girar, eliminar	Girar, eliminar	Eliminar	Eliminar ³
Idioma	Anglès	Varis	Varis	Anglès	Anglès ⁴
L. Prog.	Python	Java	Mono	Java	Java
Llicència	GPLv2	GPLv2	GPLv2	AGPL	GPL ⁵

ANDRe pretén ser una aplicació similar a PdfMod, però amb algunes diferències destacables:

- Previsualització en temps real: a mida que l'usuari arrossegui el *thumbnail* d'una pàgina, la resta de *thumbnails* es mouràn permetent veure de forma contínua el resultat.
- Llenguatge de programació: ANDRe estarà programat en Java.
- Compatibilitat amb documents: tot i que al principi ANDRe només serà compatible amb documents PDF, la seva estructura permetrà afegir nous formats *fàcilment*.

1.4 Planificació temporal

Obviant la realització de la memòria del projecte, les tasques a realitzar, acompanyades d'una estimació en hores, són les següents:

¹No és prioritari.

²No és prioritari.

³No es descarta afegir altres eines d'edició.

⁴Serà l'idioma inicial, però preparar el programa per a ser traduït sense modificar el codi font és una prioritat.

⁵Versió per determinar després d'escollir les biblioteques que s'usaran.

1. Publicar el projecte en un lloc de desenvolupament, configurant el sistema de control de versions, la llista de correu, els fòrums i el sistema de seguiment d'errors. [7h]
2. Estudiar i escollir les biblioteques. [20h]
3. Escollir la llicència. [7h]
4. Realitzar un esquema del disseny del programa. [20h]
5. Fase de disseny i desenvolupament 1: capa d'abstracció de lectura i creació de documents. [26h]
6. Fase de disseny i desenvolupament 2: lectura i creació de documents PDF. [32h]
7. Fase de disseny i desenvolupament 3: creació de la interfície d'usuari. [120h]
8. Alliberar el primer codi i escriure instruccions per a començar a desenvolupar (pels possibles col·laboradors). [6h]
9. Escriure la documentació d'usuari, les instruccions per a traduir l'aplicació i afegir captures de pantalla al lloc web. [14h]

Els punts 5 a 7 no separen disseny de desenvolupament perquè el disseny podrà estar influenciat (o retroalimentat) pels resultats que es vagin veient en el desenvolupament. L'últim punt aglutina tres tasques que, per ser petites i estar orientades a l'usuari final, es consideren com una sola tasca en aquesta planificació.

Les dependències entre les tasques descrites són:

- $3 \leftarrow 2$
- $9 \leftarrow 7 \leftarrow 6 \leftarrow 5 \leftarrow 4$
- $8 \leftarrow (1, 3 \text{ i començar } 5)$

on $a \leftarrow b$ indica que cal acabar b per a començar a . En l'última dependència, no cal acabar la cinquena tasca (però si haver-la començat) per a començar la vuitena.

L'ordre i les dependències de les tasques no són estrictes, ja que és molt probable que durant la realització del projecte calgui refer parcial o totalment una tasca que s'havia donat per acabada.

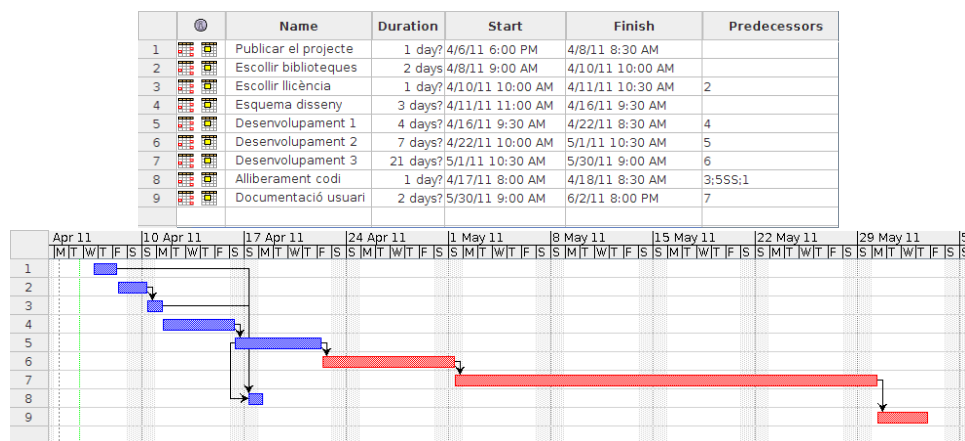


Figura 1.1: Diagrama de Gantt

1.4.1 Fites temporals

Per a la realització d'ANDRe, s'han marcat les següents tres fites temporals:

1. 15/05/2011 Disseny i desenvolupament de la capa interna del programa (fases 1 i 2 de disseny i desenvolupament) i publicació del primer codi. Correspon a les tasques 1 a 6 i 8, amb una durada total estimada de 86 hores.
2. 02/05/2011 Disseny i desenvolupament de la interfície d'usuari. Correspon a la tasca 7, amb una durada estimada de 120 hores.
3. 03/05/2011 Documentació, traducció i captures de pantalla. Correspon a la tasca 9, amb una durada estimada de 14 hores.

L'última fita està fixada 6 dies abans de la data prevista de finalització del projecte amb previsió de possibles retrassos o aparició de problemes imprevistos.

1.4.2 Diagrama de Gantt

A la figura 1.1 s'exposen les tasques descrites en un diagrama de Gantt. Les barres temporals poden tenir una longitud no proporcional a les hores de treball que representen, perquè les hores de treball estan repartides de forma no homogènia durant la durada del projecte.

1.5 Estructura de la memòria

Aquesta memòria està estructurada en cinc capítols. En el capítol 1 s'exposen els objectius que es pretenen assolir amb la realització d'aquest projecte, es fa un anàlisi comparatiu d'algunes aplicacions similars a ANDRe i s'acaba fent una planificació temporal pel projecte.

En el capítol 2 s'analitzen algunes de les eines que s'utilitzen en la gestió de projectes de programari i que s'han fet servir per a aquest projecte: sistemes de control de versions i de seguiment d'errors, i llistes de correu. Acaba amb un petit resum sobre com gestionar un projecte de programari lliure per tal d'atraure a la comunitat, tant d'usuaris com de desenvolupadors.

Al capítol 3 s'exposen algunes de les eines utilitzades en la realització del projecte. Tot seguit es resumeix l'estructura del codi del programa i els resultats dels tests realitzats, i s'explica com escriure traduccions de l'aplicació. El capítol es tanca amb una breu exposició de l'etapa de publicació del projecte.

Els resultats obtinguts s'exposen en el capítol 4, mitjançant una explicació de l'ús del programa acompanyada de captures de pantalla.

Finalment, s'avalua l'assoliment dels objectius en el capítol 5, on també es fa una breu exposició de l'estratègia que hauria de seguir el projecte en el futur. El capítol acaba amb una petita avaluació personal de l'experiència de realitzar aquest projecte.

Capítol 2

Eines de gestió de projectes¹

El desenvolupament de programari lliure es caracteritza, en part, per ser un desenvolupament col·laboratiu: diverses persones, comunicades sovint només a través d'Internet (o d'altres xarxes), treballen en les diferents tasques de les que es compona el projecte. La realització eficient d'un projecte d'aquesta mena implica la necessitat d'una sèrie d'eines per a facilitar la comunicació i el treball en grup. Per tot això, es diu que està orientat a la xarxa i a les eines que el suporten.

Algunes de les eines utilitzades en el desenvolupament de programari lliure són:

- Fòrums i llistes de correu, per a la comunicació entre integrants del projecte (desenvolupadors i usuaris).
- Sistema de control de versions, especialment útil en projectes amb més d'un desenvolupador.
- Sistema de seguiment d'errors, que permet, entre altres coses, que els errors trobats pels usuaris siguin assignats automàticament als desenvolupadors.
- Eines d'administració, com base de dades, interfície web i línia de comandes.

La instal·lació, configuració i manteniment d'aquestes i altres eines suposa una gran càrrega de treball. Per a projectes grans amb molts recursos això pot no ser un problema, però per a projectes més petits és molt recomanable utilitzar llocs de desenvolupament, que són serveis en xarxa d'algunes empreses i organitzacions que ofereixen aquestes eines.

Entre els diferents llocs de desenvolupament cal destacar, per la seva popularitat, BerliOS i, sobre tot, SourceForge. Aquest últim ofereix una

¹Aquest capítol és un resum dels apunts de l'assignatura Desenvolupament de projectes de programari lliure de la UOC, que es poden obtenir a [11]

completa interfície web des de la qual podem crear i administrar els nostres projectes. L'administració inclou el sistema de control de versions (s'ofereix, entre d'altres, CVS), sistema de seguiment d'errors i llistes de correu. SourceForge també posa a la nostra disposició un accés amb línia de comandes (SSH) per a poder administrar el CVS i el lloc web del projecte. Cal esmentar que l'anglès és l'únic idioma disponible en aquest lloc de desenvolupament, que la creació de nous projectes no és automàtica, sinó que està moderada pels administradors i que és imprescindible que els projectes tinguin una llicència lliure per a ser acceptats.

2.1 Control de versions

Durant el procés de desenvolupament de programari és desitjable emmagatzemar les diferents versions que es vagin produïnt d'un mateix fitxer. També hauria de ser possible definir i fusionar diferents branques de desenvolupament. Si, a més, hi ha diversos programadors involucrats en un projecte, és necessari poder gestionar el treball de cadascun d'ells. Les eines de control de versions faciliten i automatitzen aquestes tasques, i poden servir per a publicar el codi mitjançant un accés de només lectura obert a tothom.

2.1.1 CVS

Una de les eines de control de versions més coneguda i utilitzada és CVS (Concurrent Versions System). Es tracta d'un sistema força antic amb certes limitacions, com ara que només pot fer un seguiment de fitxers de text i no pot mantenir versions de directoris. Aquestes i d'altres limitacions estan superades en un altre sistema, Subversion, que també és molt popular.

CVS està inclòs per defecte en moltes distribucions GNU/Linux, de manera que es pot instal·lar molt fàcilment. Una vegada instal·lat, cal configurar el sistema afegint-hi comptes d'usuari (desenvolupadors) i, molt recomanable en un projecte de programari lliure, permetent-hi l'accés anònim. Pel primer cas, cal tenir un usuari al sistema operatiu per cada desenvolupador i posar tots aquests usuaris en un grup comú (per exemple, *src*). En el cas de l'accés anònim, la configuració es fa a nivell de CVS fent servir el protocol *pserver*.

Operacions bàsiques

Les operacions bàsiques de CVS són: creació d'un projecte, obtenció d'un projecte (*checkout*, creació d'una còpia local per poder-hi treballar), creació de fitxers i directoris, modificació de fitxers i fusió dels canvis, actualització de la còpia local (*update*) i eliminació de fitxers. La modificació de fitxers i fusió dels canvis (*check in* o *commit*, que assigna un nou número de versió als fitxers modificats) mereix una atenció especial, ja que és l'operació més

delicada i una de les raons de ser dels sistemes de control de versions. En CVS, el *check out* d'un fitxer per part d'un desenvolupador no bloqueja aquest fitxer, de manera que diversos desenvolupadors poden estar treballant en la seva còpia local d'un mateix fitxer alhora. Quan un d'ells vol realitzar un *check in* del fitxer, pot passar que el CVS no pugui fusionar els canvis de forma automàtica si, per exemple, els canvis que s'intenten fusionar entren en conflicte amb canvis realitzats per un altre usuari. En aquest cas, el conflicte s'haurà de resoldre manualment.

Etiquetes i branques

Un projecte pot estar format per una gran quantitat de fitxers i, cadascun d'ells, pot tenir un número de versió diferent, ja que aquests números s'assignen de forma independent per a cada fitxer. Assignant una mateixa etiqueta a una versió de cadascun dels fitxers, estem definint una versió o estat de tot el projecte. Això és útil, per exemple, per a realitzar versions beta del projecte.

Una vegada es publica el projecte, cal mantenir el codi tot solucionant els errors que els usuaris vagin trobant. D'altra banda, sovint es voldrà continuar el desenvolupament del projecte per tal d'incloure-hi noves funcionalitats per a una nova versió. Això planteja la necessitat de tenir dos desenvolupaments en paral·lel: un per corregir errors i un altre per afegir funcionalitats. El CVS ofereix aquesta possibilitat mitjançant les branques de desenvolupament, que permeten fer diversos desenvolupaments paral·lels i fusionar-los més endavant.

2.2 Sistemes de seguiment d'errors

Un sistema de seguiment d'errors o BTS (*bug tracking system*) és una eina que facilita la notificació d'errors en el programari, l'assignació d'aquests als programadors corresponents i, en general, la gestió de tota la informació referent als errors. Un BTS ha de poder emmagatzemar un històric d'errors, assignar diversos atributs a cada error, notificar automàticament dels canvis d'estat a les parts interessades (per exemple, als programadors sobre el descobriment d'un error nou, o als usuaris que han descobert l'error sobre la correcció d'aquest) i generar estadístiques d'error. A més, ha d'estar dissenyat com un sistema multiusuari amb assignació de permisos.

2.2.1 Bugzilla

Un dels BTS lliures més utilitzats és Bugzilla. És un sistema fàcil de gestionar mitjançant la seva interfície web, tot i que també es pot gestionar des de la línia d'ordres.

Els errors en Bugzilla s'organitzen en *productes* i *components*. Cada producte consta de diversos components als quals corresponen els errors. Normalment s'associa un producte de Bugzilla a un projecte de programari i els components corresponen a les parts d'aquest projecte. Cada component té un propietari, a qui inicialment s'assignen tots els errors del component.

Per a instal·lar Bugzilla (per exemple, sota Debian), caldrà tenir primer instal·lat i configurat MySQL, que és el sistema gestor de bases de dades que Bugzilla farà servir. Una vegada instal·lat, caldrà configurar-lo. Això es pot fer des de la interfície web que ofereix. Des d'aquesta interfície es poden crear i configurar usuaris i grups (més endavant, es pot deixar que els usuaris anònims obrin els seus propis comptes d'usuari).

També des de la interfície web, els usuaris podran notificar els errors que vagin trobant al projecte. Per això primer hauran d'identificar a quin producte i component correspon l'error, donar dades addicionals de l'entorn en què s'ha produït l'error, mirar d'explicar com es pot reproduir i resumir-lo.

Bugzilla permet també fer cerca d'errors. És molt important fer aquestes cerques abans de notificar errors, per a evitar així notifikacions duplicades. Una vegada localitzat un error, se'n pot consultar l'estat, que pot ser: nou, assignat, resolt, tancat (verificat) o reobert (nou). Resolt no vol dir necessàriament corregit, sinó que també pot significar que l'error és duplicat, invàlid (no hi ha error) o que no es corregirà. Un error resolt o tancat pot passar a ser reobert o nou si es detecta que estava mal resolt. A part del canvi d'estat, es pot reassignar un error, canviar-ne les dades bàsiques i afegir-hi comentaris.

2.3 Llistes de correu electrònic

Una llista de correu és una eina de comunicació molt utilitzada en les comunitats virtuals. Bàsicament, consisteix en una aplicació que emmagatzema les adreces de correu electrònic dels subscriptors (membres que volen estar comunicats) i ofereix una adreça de correu, de manera que els missatges enviats a aquesta adreça són rebuts automàticament pels subscriptors. En funció de l'aplicació i de la configuració, pot oferir altres característiques, que s'anomenaran en el cas de l'aplicació *Mailman*.

Durant el desenvolupament d'un projecte de programari lliure, la necessitat de tenir una eina de comunicació sorgeix en diverses ocasions: al principi, per comunicar els desenvolupadors entre sí, després de publicar l'aplicació, per facilitar la comunicació entre els usuaris i per obtenir *feedback* d'aquests, o cada vegada que es plantegi una tasca a realitzar de forma col·laborativa. Aquestes necessitats de comunicació es poden cobrir amb llistes de correu o bé amb altres solucions com fòrums, grups USENET o solucions mixtes.

2.3.1 Mailman

Mailman és una aplicació lliure de llistes de correu. Està inclòs en algunes distribucions de GNU/Linux com ara Debian, de manera que es pot instal·lar fàcilment des dels paquets de la distribució. Una vegada instal·lat Mailman, es poden crear i configurar les llistes de correu des de la línia de comandes i editant manualment algun fitxer (normalment `/etc/aliases`). La primera llista que s'ha de crear, anomenada *mailman*, és la llista per a gestionar el propi sistema. Les llistes es poden configurar mitjançant una interfície web. Entre les opcions de configuració n'hi ha referents a seguretat (contrassenyes per als usuaris administradors o moderadors) i a la privadesa (si la llista es fa pública o no).

Operativa d'usuaris

Un usuari pot passar a formar part d'una llista si és convidat i accepta la invitació, si hi és afegit i no s'hi esborra o si s'hi apunta per iniciativa pròpia. Segons les característiques de la llista, la última opció pot no ser possible. També pot donar-se de baixa de la llista. En qualsevol cas, la interacció amb la llista es pot fer des d'una interfície web o simplement per missatges de correu electrònic.

Per a enviar missatges a la llista, n'hi ha prou d'enviar un e-mail a l'adreça oferta per aquesta. Segons la configuració de la llista pot ser necessari que l'usuari estigui registrat o que el missatge hagi de ser autoritzat per un moderador abans de ser reenviat a la resta d'usuaris.

L'usuari també pot utilitzar la interfície web per consultar l'arxiu de missatges o per canviar la seva configuració personal, que inclou triar si vol o no rebre els missatges i si vol rebre'ls un per un o resumits en blocs (mode *digest*).

Operativa d'administració

L'administrador pot donar d'alta i de baixa llistes de correu mitjançant la línia de comandes. A més, per a cada llista de correu, pot modificar totes les opcions de configuració (opcions generals, de privadesa, de moderació, etc.) i fer tasques administratives (per exemple, subscriptions, invitacions o baixes d'usuaris) mitjançant una interfície web.

2.4 La gestió d'un projecte

L'èxit d'un projecte de programari lliure no es pot garantir únicament amb un bon programa. Per definició de programari lliure, un projecte només serà exitós si s'aconsegueix la implicació de la comunitat fins al punt que aquesta resolgui per si sola les necessitats que es planteja. Les eines plantejades en

els altres apartats són un pas necessari, però no suficient, per a aconseguir aquesta implicació. De fet, cal dedicar un gran esforç a la gestió de recursos i de la comunitat, sense els quals el projecte segurament fracasarà.

2.4.1 Llicència

L'últim pas abans de fer públic un projecte, és triar-ne la llicència. És un pas important que té influència en el desenvolupament posterior del projecte i que en marca un punt d'inflexió perquè, una vegada publicat amb una llicència lliure, els desenvolupadors inicials del projecte hauran de treballar-hi com si el projecte ja no els hi pertanyés.

2.4.2 Atraure la comunitat

Com en qualsevol empresa o producte, la imatge és molt important. I en un programa, la imatge és sovint el seu web. Per això cal que el projecte tingui un web ben estructurat, que sigui actualitzat sovint, que ofereixi informació clara del projecte, un apartat de descàrregues amb diferents opcions (paquets per a diferents distribucions, ja que és molt important que el programa sigui fàcil d'instal·lar, juntament amb el codi font), un apartat de notícies i un de documentació. És molt recomanable incloure-hi tutorials i captures de pantalla, per tal de fer els primers passos el més senzill possible. Per a aconseguir la implicació dels usuaris, el web hauria d'incloure enllaços a les eines de comunicació (l·listes de correu, fòrums), eines de seguiment d'errors i de control de versions, així com explicacions de les configuracions necessàries per a utilitzar-les.

No menys important és intentar cridar l'atenció cap al projecte publicitant-lo en la xarxa. Per a això es poden fer servir pàgines especialitzades en programari lliure i blogs.

Per a facilitar la col·laboració de desenvolupadors, cal que el codi font del programa sigui clar i estigui ben estructurat. Utilitzar l'anglès en tota la informació referent al projecte també pot ajudar a aconseguir col·laboradors d'arreu del món, però cal estar preparats per la internacionalització en funció del públic objectiu del programa.

Capítol 3

Realització del projecte

3.1 Programari, llenguatges i serveis utilitzats

3.1.1 Java

Java és una plataforma de programari desenvolupada per Sun Microsystems (ara Oracle) que possibilita el desenvolupament de programari multiplataforma. Bàsicament consta d'una màquina virtual, diferent en cada combinació de maquinari i sistema operatiu i sobre la qual s'executen les aplicacions de Java, d'un extens conjunt de biblioteques estàndar, d'un conjunt d'eines de desenvolupament i d'un llenguatge de programació, que també s'anomena Java. Gairebé tot el codi de Java està publicat sota llicència GPL.

Javadoc

És una de les eines de desenvolupament incloses en Java. Serveix per a generar la documentació en format *html* de les Interfícies de Programació d'Aplicacions (*Application Programming Interface*, o *API*, en anglès) automàticament a partir de comentaris escrits en el codi amb una sintaxi especial[12].

3.1.2 Eclipse[13]

És un entorn integrat de desenvolupament (*IDE*, per les seves inicials en anglès) desenvolupat inicialment per IBM i després per la Eclipse Foundation. És molt flexible i està pensat per ser ampliat amb mòduls diversos. Cal destacar que disposa d'un client de CVS integrat per defecte. Juntament amb NetBeans, és un dels *IDEs* més populars per a programar en Java. Està programat en Java i és lliure.

3.1.3 SourceForge

És un dels llocs web de desenvolupament més populars. Ofereix les eines de col·laboració entre desenvolupadors necessàries per a complir els objectius del projecte (sistema de control de versions, llistes de correu, fòrum, sistema de seguiment d'errors, lloc web). Un dels sistemes de control de versions que ofereix és el CVS, que és el que s'utilitzarà, en combinació amb el client integrat en Eclipse, durant el desenvolupament d'ANDRe.

3.1.4 L^AT_EX[14]

És un llenguatge de marques per a documents (*document markup language*, en anglès) que permet escriure documents centrant-se en el contingut i no en la forma. Les marques s'utilitzen per a especificar l'estructura lògica del document, que serà formatat per un programa que les interpretarà.

Molt usat en la comunitat científica i matemàtica (universitats, llibres i revistes), és el llenguatge emprat per a escriure aquesta memòria.

3.2 El·lecció de les biblioteques

Algunes de les biblioteques que s'han considerat són: iText, Pdf-renderer, jPod, jPodRenderer, JPedal i PDFBox per a la manipulació de PDFs i Swing, MiGLayout i biblioteques de JGoodies per a la interfície gràfica. Swing forma part de les biblioteques estàndard de Java, mentre que les altres formen part d'altres projectes i es resumeixen a continuació:

iText[15] Biblioteca per a manipular i generar documents PDF. Disponible amb llicències GNU AGPLv3 i propietària.

PDF-Renderer[16] Biblioteca per a visualitzar documents PDF. Distribuïda sota la llicència GNU LGPLv2.1.

jPod[17] i jPodRenderer[18] Biblioteques per a manipular i visualitzar documents PDF. Disponibles sota les llicències BSD i GNU GPL respectivament.

JPedal[19] Biblioteca per a visualitzar documents PDF. Disponible sota una llicència GNU LGPL.

PDFBox[1] Biblioteca de l'Apache Software Foundation per a crear i manipular documents PDF. Publicada sota la llicència Apache License v2.0[20].

MiGLayout[21] Potent biblioteca per a ajudar en la distribució de components en una interfície gràfica creada amb Swing. Distribuïda sota llicències BSD i GNU GPL.

JGoodies[22] Conjunt de biblioteques amb llicència BSD destinades a millorar la interfície gràfica de programes escrits en Java que utilitzen Swing.

D'entre les biblioteques de manipulació de PDFs, PDFBox ha estat la primera el·lecció per ser la única que cobreix les necessitats tant de crear PDFs com de generar imatges a partir de les pàgines d'un PDF. El fet d'estar realitzada per una organització amb renom com l'Apache Foundation també ha influït positivament en la decisió. Després de provar-la amb diversos documents, s'ha observat que:

- No presenta problemes a l'hora de generar documents.
- En la generació d'imatges a partir de les pàgines de documents, presenta petits problemes quan aquests contenen imatges vectorials. Les imatges generades no reproduïxen amb fidelitat el contingut de les pàgines, però permeten igualment identificar-les. Es tracta d'un problema menor.
- La velocitat de generació d'imatges és, subjectivament, baixa. És un punt a millorar.

Amb l'objectiu de millorar el tercer punt, ha estat provada la biblioteca Pdf-Renderer. Aquesta ha demostrat ser sensiblement més ràpida alhora de generar imatges, però té una mancança inacceptable: les imatges generades no mostren les imatges en format *jpeg* contingudes en les pàgines. Aquest fet ha fet descartar aquesta biblioteca.

Per a poder dedicar els recursos temporals en les altres etapes del projecte, i tenint en compte que l'estructura del programa ha de permetre canviar amb relativa facilitat les biblioteques utilitzades, s'ha decidit utilitzar únicament PDFBox i deixar possibles optimitzacions en aquest aspecte per a futures versions del programa.

Pel què fa a les biblioteques per a realitzar interfícies gràfiques, ha estat suficient l'ús de Swing. Per a aconseguir que els *thumbnails* es moguin de manera suau, ha estat necessari controlar-ne la posició de forma directa, sense recórrer a cap *Layout Manager* com MigLayout.

3.3 El·lecció de la llicència

La llicència escollida per a publicar ANDRe és la *GNU General Public License Version 3*. Els motius d'escollir una llicència GPL són d'una banda, una inclinació personal envers les llicències de programari lliure robustes, i de l'altra, el fet de tractar-se de llicències ben conegudes. L'el·lecció de la versió 3 és deguda a que és la única versió de les GPL que és compatible amb la llicència *Apache License 2.0*[23].

3.4 Fases del desenvolupament del programa

En aquesta secció es comenten els aspectes més rellevants de les tres fases de disseny i implementació del programa exposades a 1.4. En les explicacions s'han usat alguns conceptes bàsics de la programació orientada a objectes, com *classe* i *interfície*, i notació bàsica pròpia de Java, però que resultarà evident a qualsevol lector familiaritzat amb el desenvolupament d'aplicacions.

3.4.1 Capa abstracta de lectura i creació de documents

En aquesta fase es pretenia definir les entitats abstractes necessàries per a manipular documents a nivell de pàgina. Havia de ser possible separar un document en pàgines, crear un document nou a partir de pàgines i convertir pàgines en imatges. Semblava obvi, doncs, que s'haurien de definir les entitats *document* i *pàgina*.

Document

El programa ha estat dissenyat per a crear nous documents partint d'altres documents, però no per a editar-ne de ja existents, fet que garanteix que no es podrà perdre cap pàgina. Aquesta diferenciació entre documents d'origen (que no es modificaran) i document de destinació (que s'editarà), es veu reflexada en la creació d'una interfície diferent per a cada cas:

Interfície `ImmutableDocument` És la interfície que hauran d'implementar les classes destinades a representar documents d'origen. Consta dels següents mètodes, tots ells públics:

- `open():void` Carrega un document desat en un fitxer. El nom del fitxer no es passa com a paràmetre d'aquest mètode, sinó que ha d'estar associat a cada instància de `ImmutableDocument` (per exemple, passant-lo en el mètode constructor). D'aquesta manera s'evita carregar diferents fitxers des d'una mateixa instància.
- `close():void` Allibera els recursos del sistema utilitzats (normalment, hauria de tancar el fitxer obert a `open()`).
- `getType():DocumentType` Identifica de quin tipus o format de document es tracta. Serà útil en el futur si ANDRe arriba a suportar altres formats a part del PDF. `DocumentType` és una classe que simplement enumera els diferents formats acceptats.
- `getPages():List<Page>` Retorna una llista ordenada de les pàgines contingudes en el document. Les pàgines estan representades amb objectes d'una classe que implementa la interfície `Page`, que es resumirà més endavant.

- `getName():String` Retorna un nom que identifica inequívocament el document (per exemple, el nom del fitxer associat amb la ruta completa de directoris). És útil per a mostrar informació sobre les activitats del programa en el terminal.
- `getShortName:String` Retorna un nom curt associat al document (per exemple, el nom del fitxer associat, sense ruta de directoris). Serà el nom que es mostrarà en la interfície gràfica.

Interfície `EditableDocument` És la interfície que hauran d'implementar les classes destinades a representar documents de destinació. Consta dels següents mètodes, tots ells públics:

- `insertPage(Page page):void` Afegeix una pàgina al final del document.
- `insertPage(Page page, int index):void` Afegeix una pàgina en la posició indicada del document.
- `movePage(int originIndex, int destinationIndex):void` Canvia una pàgina de posició.
- `removePage(int index):void` Esborra la pàgina situada en la posició indicada.
- `saveInFile(String fileName):boolean` Desa el document en el fitxer indicat, en un format determinat per la classe que implementa aquesta interfície. Retorna *cert* si, i només si, s'ha pogut desar el document i tancar el fitxer.

Pàgina

En aquest cas només ha estat necessari definir una interfície que serveix, bàsicament, per a identificar cada pàgina dins del seu document d'origen i per a convertir-les en imatges:

Interfície `Page` Els mètodes definits en aquesta interfície, tots públics, són:

- `getOriginalDocument():ImmutableDocument` Identifica el document d'origen de la pàgina.
- `getOriginalPageNumber():int` Retorna la posició de la pàgina en el seu document d'origen.

- `getThumbnail() : BufferedImage` Converteix la pàgina en imatge, la mida de la qual està determinada per constants globals del programa. L'objecte que retorna és de tipus `BufferedImage`, que és una classe inclosa en Java.
- `getPageAs(DocumentType type) : Object` Retorna la pàgina en un objecte el tipus del qual depèn del format de document passat com a paràmetre i de les biblioteques utilitzades per a aquest format. Per exemple, si es passa *PDF* com a paràmetre, i es fa servir *PDFBox*, retorna un objecte de tipus `PDPPage`, que és la classe que fa servir *PDFBox* per a representar pàgines.

3.4.2 Lectura i creació de documents PDF

En aquesta etapa s'han creat tres classes que implementen les tres interfícies definides en l'etapa anterior per al format de document PDF. Aquestes classes es basen en classes de la biblioteca *PDFBox*, que ja implementen mètodes similars als definits en les interfícies. Aquesta fase ha mostrat algunes dificultats degudes a la manca de documentació de qualitat i d'exemples d'ús de *PDFBox*.

Classes **ImmutablePdf** i **EditablePdf**

Implementen les interfícies `ImmutableDocument` i `EditableDocument`, respectivament. Es basen en la classe `PDDocument` de *PDFBox*, que és la classe que utilitza aquesta biblioteca per a representar documents PDF.

Classe **PdfPage**

Implementa la interfície `Page`. Es basa en la classe `PDPPage` de *PDFBox*.

3.4.3 Interfície gràfica d'usuari

Ha estat la fase més llarga i la que ha presentat més dificultats. A més, també es pot considerar la fase amb més potencial de millora. Fa un ús intensiu de fils d'execució (execució d'ordres en paral·lel), utilitza la biblioteca *Swing* de Java a un nivell força baix i implementa suport per a *Drag and Drop*. Gairebé tot el codi font corresponent a aquesta etapa pertany a dos paquets (i els seus subpaquets), que són:

Paquet **gui**

Conté les classes encarregades de representar la finestra de l'aplicació i tots els seus elements gràfics. Cal destacar-ne les classes:

- `FloatingPanel` Panell que pot contenir altres elements gràfics i amb capacitat de desplaçar-se suaument per a ocupar una posició donada. Els desplaçaments estan controlats per instàncies de la classe `FloatingPanelSlider` associades a cada instància d'aquesta classe.
- `FloatingPanelSlider` Fil d'execució que controla el desplaçament dels `FloatingPanel`. Per a fer-ho, calcula a intervals fixats de temps la posició que hauria d'ocupar el panell, genera un esdeveniment de tipus `FloatingPanelLocateEvent` i afegeix aquest esdeveniment a la cua d'esdeveniments de *Swing*.
- `FloatingPanelLocateEvent` Esdeveniment que ordena a *Swing* el desplaçament d'un panell.
- `FloatingPanelContainer` Panell encarregat de contenir objectes del tipus `FloatingPanel` en formació de graella i de controlar-ne la posició.

Les classes `FloatingThumbnail` i `FloatingThumbnailContainer`, que són extensions de la primera i última classes descrites, respectivament, implementen funcionalitats específiques per a la representació de *thumbnails*.

Paquet controller

Conté les classes que controlen les accions associades als menús de l'aplicació i al *Drag and Drop*. Cada opció dels menús té una classe associada (`About`, `OpenFile`, `Quit`, `RemovePage`, `SaveFile`), i el conjunt d'aquestes classes està recolzat amb algunes classes auxiliars (`ActionBase`, `ActionsConstants`). Les classes `DragController` i `DropController` controlen les accions d'arrossegar i deixar anar, respectivament.

3.5 Proves

ANDRe 0.1 Alpha ha estat provat amb diversos documents i sota diverses plataformes (*Ubuntu 10.04* 32 i 64 bits, *Windows XP* 32 bits i *Windows 7* 32 bits). En general, les proves han resultat satisfactòries si es té en compte que es tracta d'un programa en una fase jove de desenvolupament. No s'han detectat diferències entre les execucions en diferents plataformes, però sí que s'han detectat errors que semblen dependre dels documents d'origen i errors no reproduïbles. També s'han identificat mancances de funcionalitats bàsiques -algunes d'elles ja eren conegudes durant el desenvolupament- que fan que encara no sigui recomanable promocionar el programa.

3.5.1 Errors detectats

- Els *thumbnails* d'alguns documents PDF no mostren el text d'aquest. És degut a que el document fa servir una o més fonts que *PDFBox* no pot convertir en imatge. Possible solució: utilitzar una biblioteca alternativa.
- Error doble: el document d'origen no s'arriba a carregar del tot i l'acció de desar el document no funciona. Aquest error s'ha trobat en un document gran que es carregava molt lentament. Després de dividir amb un altre programa el document en dues parts i carregar-les per separat, ANDRe ha funcionat sense problemes, fet que fa pensar que el segon error era conseqüència del primer. Possibles causes: la màquina virtual de Java no disposa de suficient memòria. Possibles solucions: monitoritzar contínuament l'ús de memòria i no forçar la càrrega completa dels documents d'origen.
- Al desar un document, una de les pàgines sembla ser una combinació de dues de les pàgines originals. Aquest error no s'ha pogut reproduir.
- Quan s'obre un fitxer es duen a terme dues accions: lectura del document i generació dels *thumbnails*. Mentre que la segona acció s'executa en un fil propi, permetent utilitzar el programa alhora, la primera s'executa en el fil principal. Si el fitxer és gran, això provoca que l'usuari noti que el programa no respon.

3.5.2 Mancances de funcionalitats bàsiques

- Abans de sortir de l'aplicació, alertar l'usuari si aquest no ha desat el document.
- Si es prova de desar el document a sobre d'un fitxer ja existent, informar-ne i oferir les opcions de fer-ho i de cancel·lar. Actualment el programa no permet sobreescure fitxers, però tampoc informa que el document no s'ha desat.
- Afegir automàticament l'extensió *.pdf* al nom del document que es desa. Això és important perquè *Windows* basa la detecció del tipus de fitxer en l'extensió del seu nom, i no en els *Magic Numbers*[24].
- Fer que les barres de desplaçament es desplacin més ràpid amb la rodeta del ratolí.
- Permetre seleccionar més d'un fitxer en la finestra de diàleg d'obrir documents.
- Afegir funcionalitats de *zoom* de pàgines.

- Permetre seleccionar totes les pàgines d'un document amb *Ctrl+A* i esborrar les pàgines seleccionades amb la tecla de suprimir.

3.6 Localització

ANDRe està dissenyat per a ser fàcilment traduïble, de manera que els usuaris, encara que no tinguin cap coneixement de programació, poden escriure traduccions del programa en la seva llengua. Aquestes traduccions inclouen els noms dels menús, els textos dels indicadors de funció (text que es mostra en un petit requadre quan es situa el ratolí a sobre d'un menú) i la nota d'autoria i llicència. No inclouen els textos visibles en les finestres de diàleg, la traducció dels quals és feta automàticament pel Java en cas que sigui possible.

Quan s'executa el programa, aquest comprova quina llengua fa servir el sistema i busca dins del directori `translations` el fitxer de traduccions corresponent a aquesta llengua. Si no el troba, fa servir el fitxer de traduccions en anglès americà. En la distribució d'ANDRe s'inclouen traduccions d'anglès americà, de castellà i de català.

3.6.1 Escriure una traducció

Dins el directori `translations` hi ha fitxers de text anomenats `messages_xx_XX.properties`, on `xx` és un codi d'idioma (per exemple, `ca` pel català, `es` pel castellà i `en` per l'anglès), i `XX` és un codi de país (per exemple, `ES` per Espanya i `US` pels Estats Units).

Aquests fitxers estan estructurats de manera que en cada línia hi ha el nom d'una propietat, un símbol d'igualtat i el valor d'aquesta propietat, que és el que s'ha de traduir. Per exemple, el fitxer per la traducció anglesa conté:

```
ActionsConstants.AboutDesc=About ANDRe
ActionsConstants.AboutName=About ANDRe
ActionsConstants.AboutText=ANDRe 0.1 Alpha\nCopyright
2011 Domenec Martin Martinez\n\nThis program is
licensed to you under the terms\nof the GNU General
Public License Version 3\nas published by the Free
Software Foundation.
ActionsConstants.OpenFileDesc=Open a PDF file
ActionsConstants.OpenFileName=Open
ActionsConstants.QuitDesc=Exit ANDRe
ActionsConstants.QuitName=Quit
ActionsConstants.RemoveDesc=Remove this page(s)
ActionsConstants.RemoveName=Remove
ActionsConstants.SaveFileDesc=Save document as PDF
```

```

ActionsConstants.SaveFileName=Save
MainWindow.MenuFile=File
MainWindow.MenuHelp=Help

```

Per a escriure una traducció nova cal crear un nou fitxer amb el contingut d'un dels fitxers de traducció existents però traduïnt a l'idioma desitjat els valors assignats a les propietats i desar aquest fitxer dins el directori `translations` amb un nom en funció de l'idioma de la traducció, utilitzant els codis corresponents¹.

3.7 Publicació i promoció

ANDRe forma part del web de desenvolupament SourceForge des del 3 d'abril de 2011 (veure figura 3.1). De tots els serveis que ofereix el lloc, han estat activats els serveis de control de versions amb lectura anònima (CVS), fòrums (un per a desenvolupadors, un d'ajuda per a usuaris i un de debat), llistes de correu amb Mailman (una llista per a desenvolupadors), sistema de seguiment d'errors (que també serveix per a fer suggerències de millores al programa) i mostra de captures de pantalla (integrades en la pàgina de resum) (figura 3.2).

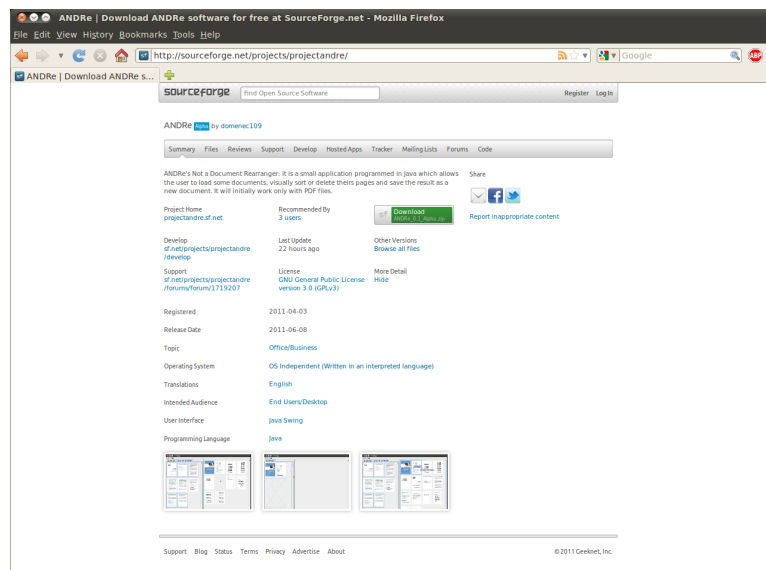


Figura 3.1: Pàgina de resum del projecte. Mostra dades bàsiques del projecte i captures de pantalla. Des d'aquí es pot accedir fàcilment a la resta de recursos de desenvolupament.

¹Els codis d'idioma i país estan disponibles al lloc web d'Oracle[26].

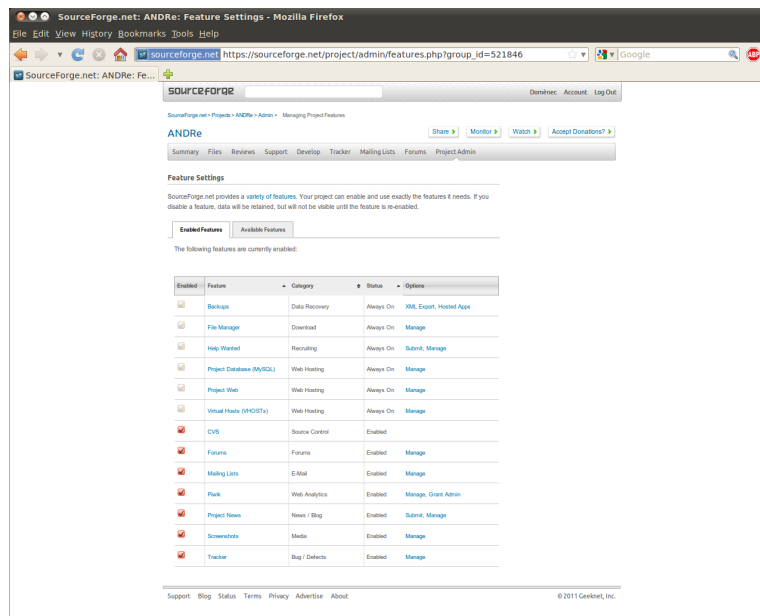


Figura 3.2: Serveis de SourceForge activats per a ANDRe.

El codi ha estat publicat contínuament durant el seu desenvolupament, fent servir així el servidor CVS per a tenir còpies de seguretat de diferents estats de desenvolupament. Fins el moment d'escriure aquesta memòria no hi ha hagut cap aportació de la comunitat en el codi, de manera que no ha estat necessari resoldre cap conflicte de versions. En la figura 3.3 es pot veure l'evolució de les operacions de lectura i escriptura en el sistema de control de versions.

Finalment, el dia 8 de juny de 2011, ha estat publicada la versió *0.1 Alpha*, disponible per a ser baixada des de la pàgina del projecte a SourceForge. Se n'ofereix el codi font, amb instruccions per a compilar-lo automàticament amb *Ant*[25], i una versió binària multiplataforma. En ambdós casos s'adjunten instruccions per a executar el programa i per a escriure'n traduccions, a més del text de la llicència.

Promoció

Considerant les mancances del programa exposades a 3.5.2, s'ha considerat oportú no promocionar el programa més enllà del cercle d'amistats, ja que fer-ho podria provocar decepció en un gran nombre d'usuaris potencials i resultar contraproduent per a l'èxit futur d'ANDRe.

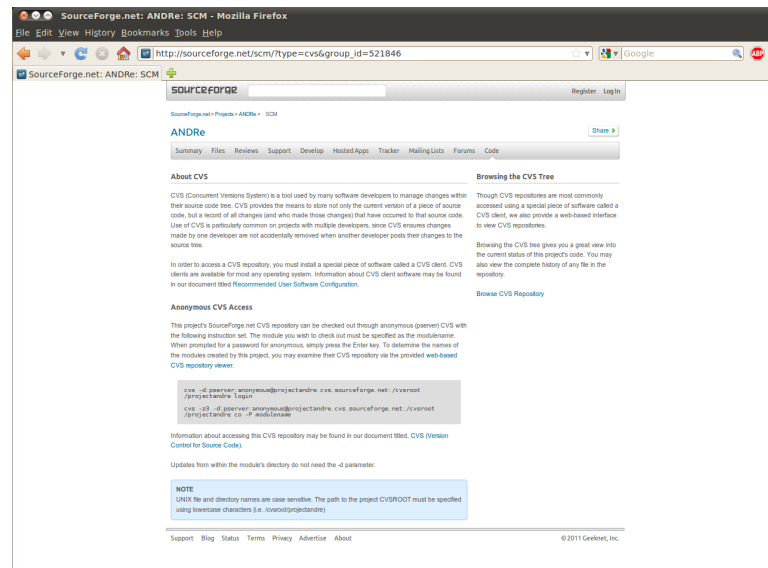


Figura 3.3: Activitat del sistema de control de versions CVS.

Capítol 4

Resultats

En aquest apartat s'explica el funcionament de la versió d'ANDRe *0.1 Alpha* a mode de manual d'usuari. Les explicacions estan complementades amb captures de pantalla que mostren, però, una versió en anglès d'aquest. La llengua en la que l'usuari veuria el programa dependria de la llengua configurada en el seu sistema i de si aquesta està disponible en les traduccions.

Aquesta versió del programa permet a l'usuari construir i desar un document nou a partir de pàgines de documents ja existents. No permet modificar directament un document ni desar utilitzant un nom d'un fitxer existent, de manera que no pot provocar pèrdua de dades.

4.1 Finestra principal

En obrir ANDRe, l'usuari veurà una finestra com la mostrada en la figura 4.1, que mostra una barra de menú a la part superior i dos espais situats de costat que ocupen pràcticament tota la finestra. El menú de fitxer ofereix opcions per a obrir un fitxer PDF, desar el document creat en PDF i sortir. El menú d'ajuda únicament presenta una opció per a obtenir informació del programa (autoria i llicència).

4.2 Obrir documents

Mitjançant el menú o la combinació de tecles *Ctrl+O* s'obre la finestra de diàleg per a obrir un document (veure figura 4.2). Aquí es pot escollir si només es volen mostrar fitxers PDF i directoris, o tots els fitxers. En qualsevol cas, el programa només permet obrir fitxers PDF, ignorant qualsevol intent d'obrir un fitxer amb un altre format o un fitxer malmès. Només es permet obrir un fitxer cada vegada que s'obre la finestra de diàleg.

Després d'obrir un document, es mostra una pestanya corresponent a aquest document a la part esquerra de la finestra (figura 4.3). En aquesta pestanya es van mostrant *thumbnails* de les pàgines del document a mida

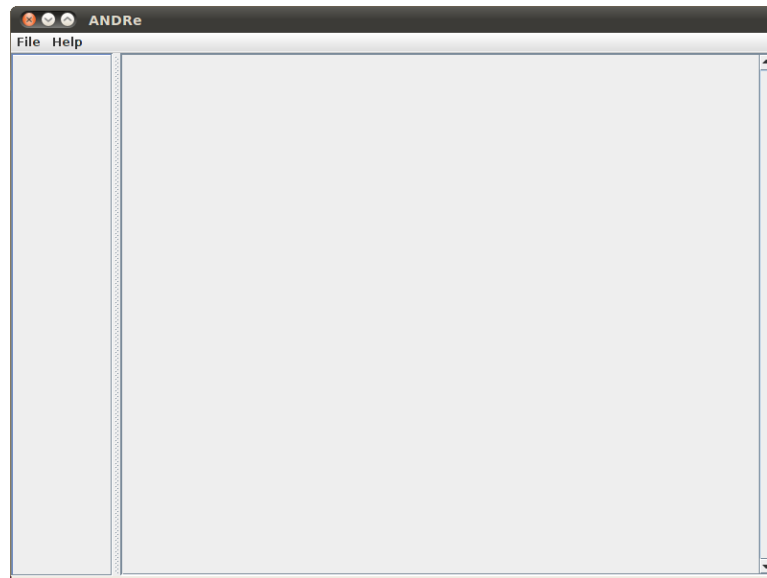


Figura 4.1: Pantalla inicial

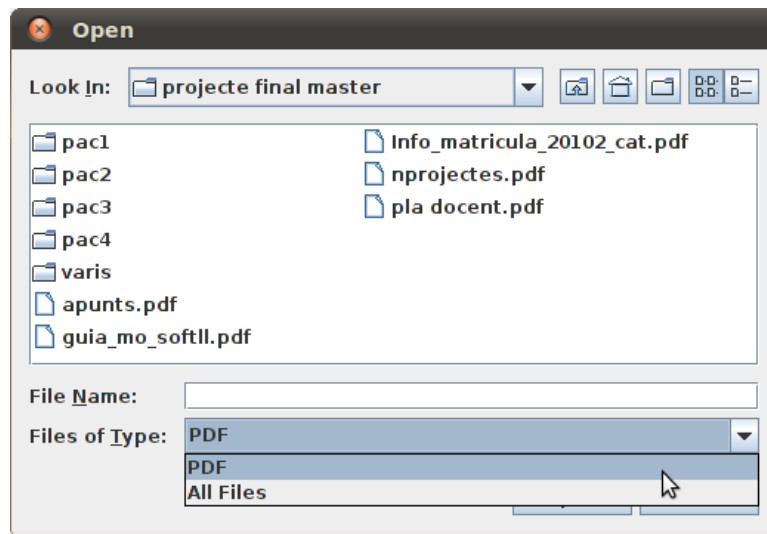


Figura 4.2: Obrint un fitxer

que aquestes es van carregant. Durant aquest procés l'usuari pot realitzar altres tasques com ara obrir un altre document o arrossegant pàgines cap a la part dreta de la finestra, encara que aquestes encara no s'hagin carregat. Desplaçar la barra vertical que separa ambdues parts de la finestra fa variar el nombre de *thumbnails* que es mostren a cada fila, que ocupen les noves

posicions tot desplaçant-se de forma suau.

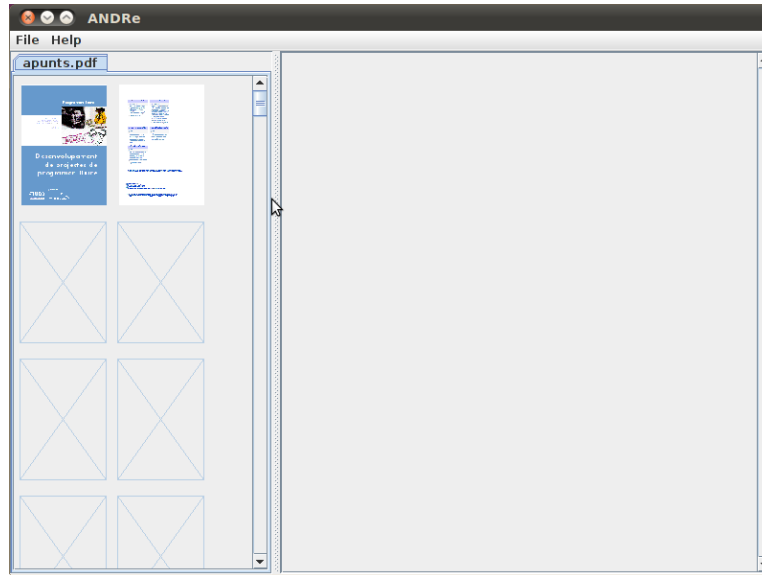


Figura 4.3: Carregant un fitxer

Si s'obren més documents, aquests es van acumulant en pestanyes a la part esquerra de la pantalla (figura 4.4). En aquesta versió del programa no és possible tancar documents.

4.3 Crear un nou document

Cada sessió del programa ANDRe permet tenir un document en edició, que ocupa la part dreta de la finestra. El nou document es crea seleccionant pàgines dels documents situats a l'esquerra i posant-les a la dreta en l'ordre desitjat.

Les pàgines es seleccionen *clickant* sobre aquestes amb el ratolí. Mitjançant les tecles *Ctrl* i *Shift* es poden fer seleccions múltiples i d'interval, incloses seleccions de múltiples intervals¹. Amb un canvi de color del marc de les pàgines s'indica si aquestes estan seleccionades (figura 4.5).

Després d'arrossegar una selecció de pàgines des d'un document font a la zona d'edició, les pàgines són visibles en el nou document però continuen estant disponibles en el document original, de manera que es poden tornar a incloure (figura 4.6).

Si en la zona d'edició ja hi ha pàgines quan s'hi afegeixen de noves, les primeres es desplacen tot *obrint pas* al punter del ratolí (figura 4.7), indicant

¹Per exemple, fent *Click 1*, *Shift + Click 4*, *Ctrl + Click 8* i *Shift + Click 10*, es seleccionen les pàgines 1 a 4 i 8 a 10.

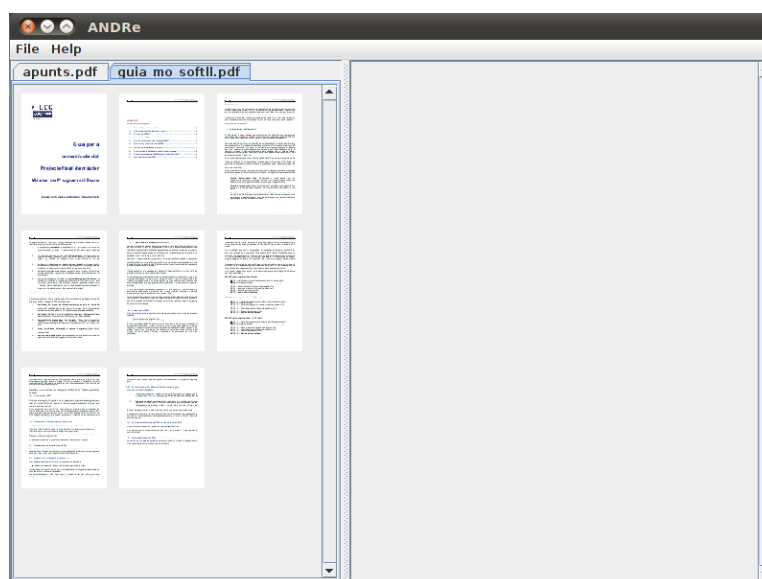


Figura 4.4: Pestanyes amb diversos fitxers oberts

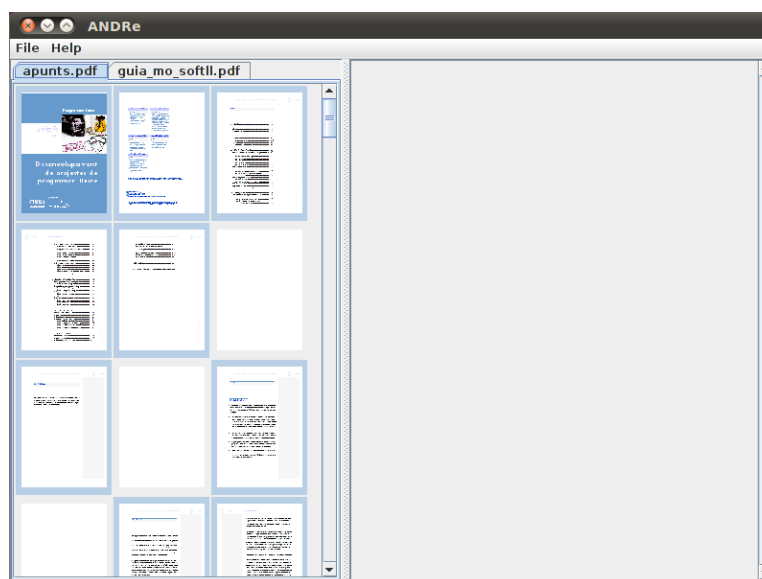


Figura 4.5: Les pàgines sel·leccionades estan marcades amb un marc blau.

en cinquena posició s'afegiran les noves pàgines (figura 4.8).

També és possible arrossegar pàgines des de la zona d'edició. En aquest cas, no se'n fan còpies, sinó que se'n canvia la posició. En el moment de començar la operació d'arrossegament, les pàgines sel·leccionades desapareixen.

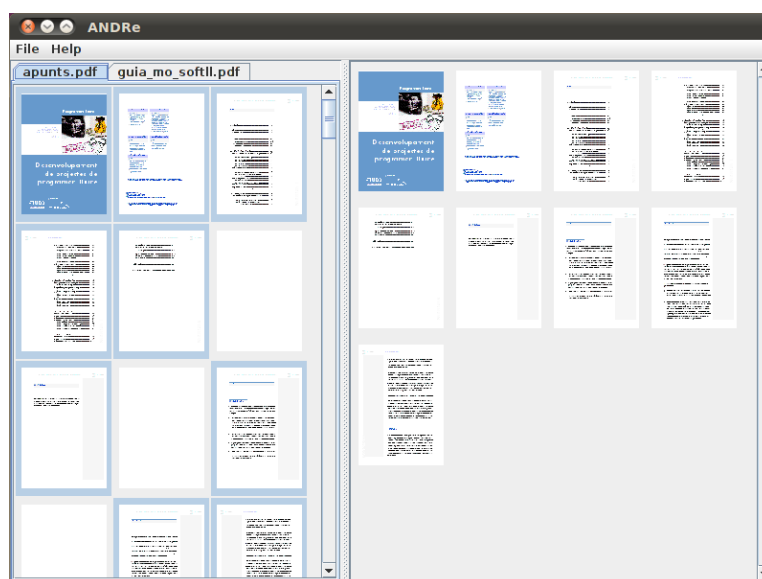


Figura 4.6: A la zona d'edició hi ha una còpia de les pàgines seleccionades al document `apunts.pdf`.

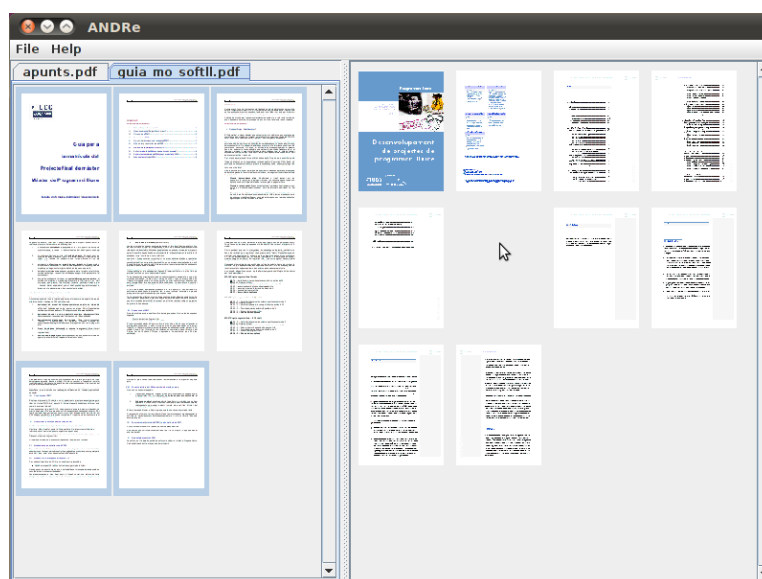


Figura 4.7: Separació entre la cinquena i sisena pàgina, indicant que les cinc pàgines que s'arrosseguen ocuparan les posicions sisena a desena.

xen. Si es deixen anar en una posició vàlida, apareixen en la nova posició. En canvi, si es deixen anar fora de la zona d'edició, tornen a aparèixer en la

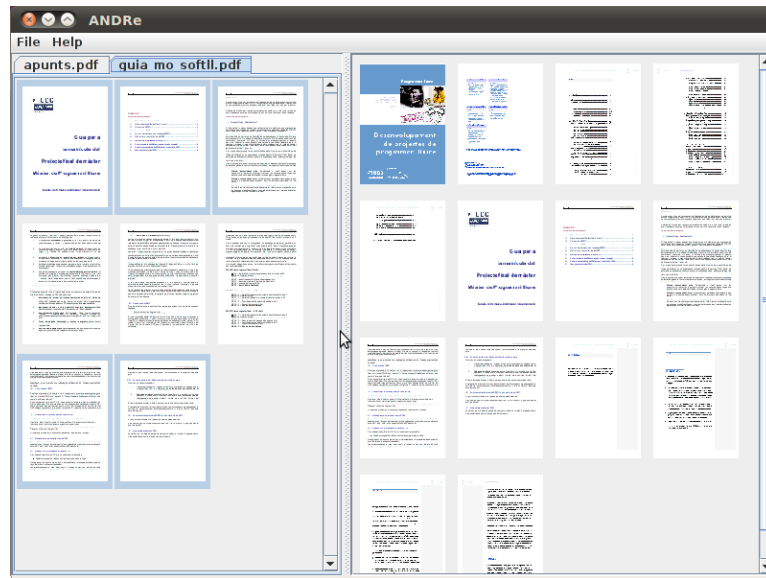


Figura 4.8: El resultat és l'esperat.

seva posició inicial.

Per a esborrar pàgines cal fer *click* amb el botó dret del ratolí sobre una pàgina o una sel·lecció de pàgines. Aleshores apareix un menú amb una única opció d'esborrar (figura 4.9). Quan s'esborren pàgines, les pàgines que romanen es desplacen suaument per a ocupar l'espai que ha quedat lliure.

Quan el nou document està llest, es pot desar en format PDF. Per a això cal obrir la finestra de diàleg de desar documents, ja sigui a través del menú de fitxer o amb la combinació de tecles *Ctrl+S*. En aquesta finestra cal escriure el nom d'un fitxer inexistent, que serà el nom del fitxer que contindrà el document desat. Si s'escull un nom d'un fitxer existent, aquesta versió d'ANDRe ignorarà l'ordre (no sobreescriurà el fitxer) però tampoc informará l'usuari d'aquest fet.

4.4 Sortir del programa

Per a sortir del programa cal escollir la opció corresponent al menú de fitxer, o bé prémer *Ctrl+Q*, o bé tancar la finestra amb el mètode ofert pel gestor de finestres (normalment, una creueta en una de les cantonades superiors de la finestra). En aquesta versió del programa no es verifica que el document editat estigui desat, de manera que qualsevol de les tres accions descrites aturarà l'execució del programa de forma immediata.

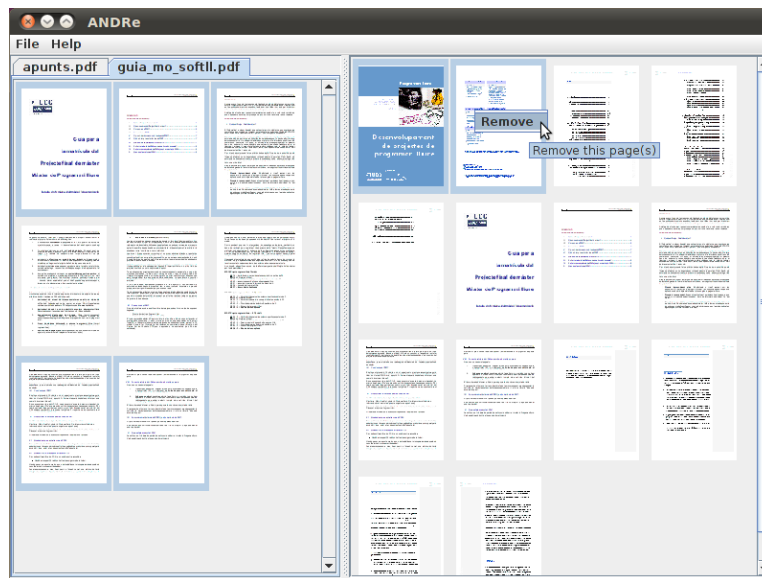


Figura 4.9: Menú contextual per a esborrar pàgines. Només es poden esborrar pàgines de la zona editable.

Capítol 5

Conclusions

5.1 Assoliment d'objectius

La versió actual d'ANDRe permet a l'usuari realitzar les tasques plantejades a 1.1 amb les limitacions intrínseques d'un programa en estat inicial de desenvolupament (*bugs*, incompatibilitat amb alguns fitxers, etc.). A més, tot el procés de desenvolupament del projecte ha estat visible per la comunitat, a la qual s'han ofert les eines necessàries per a col·laborar-hi. Per tant, es pot considerar que s'ha assolit l'objectiu general del projecte.

A continuació s'exposa un anàlisi breu de l'estat d'assoliment dels objectius específics plantejats a 1.1.1:

Objectius referents a la interfície d'usuari de l'aplicació

- Fer una interfície senzilla (fàcil d'utilitzar). Assolit.
- La interfície ha de ser intuïtiva i atractiva (desplaçament suau dels *thumbnails*). Assolit, però s'hauria de millorar mostrant què s'està arrossegant en les operacions de *Drag and Drop*.
- Oferir característiques complementàries (desfer i refer). No assolit. Les funcionalitats de fer i desfer no són imprescindibles per a una primera versió del programa, però s'haurien d'incloure en el futur si es vol garantir l'èxit d'ANDRe.

Objectius tècnics

- Escriure el codi en Java. Assolit.
- Evitar utilitzar biblioteques de plataformes específiques. Assolit: *PDFBox* està escrita en Java i és, per tant, independent de la plataforma.

- Mantenir un nivell d'abstracció que permeti, en el futur, afegir compatibilitat amb altres formats. Assolit, a través de la implementació d'una capa abstracta de lectura i creació de documents (veure 3.4.1).
- Triar una llicència lliure adequada tenint en compte les biblioteques que es volen utilitzar. Assolit: el·lecció de la llicència *GNU GPLv3*, compatible amb la *Apache License 2.0*.

Objectius d'acostament a la comunitat

- Oferir un sistema de control de versions amb possibilitat de lectura anònima. Assolit (SourceForge).
- Oferir una llista de correu destinada als desenvolupadors. Assolit (SourceForge).
- Oferir un fòrum per als desenvolupadors i un per als usuaris. Assolit (SourceForge).
- Oferir un sistema de seguiment d'errors. Assolit (SourceForge).
- Tenir un lloc web per al projecte, ben estructurat i amb elements atractius (captures de pantalla). Assolit parcialment: la pàgina de resum del projecte a SourceForge és, en el moment d'escriure aquesta memòria, el més semblant a un lloc web del projecte. Tot i que està ben estructurada i conté captures de pantalla, té un format genèric i no és, per tant, especialment atractiva.
- Escriure el codi de forma ordenada i documentar-lo bé. Assolit: el codi està estructurat en paquets i classes, i gairebé totes les classes i els seus mètodes públics estan comentats mitjançant *javadoc*.
- Explicar detalladament com obtenir el codi i quins passos s'han de seguir per a fer-lo funcionar. Assolit. A la pàgina del projecte al SourceForge, a l'apartat de descàrregues, s'ofereix el codi font amb instruccions per a compilar-lo. A més, a l'apartat de codi, s'indica com obtenir el codi directament del servidor CVS.
- Utilitzar l'anglès en les comunicacions i la documentació. Assolit.
- Fer el programa fàcilment internacionalitzable. Assolit: el programa ja ha estat traduït a dues llengües, a més de l'anglès, i s'han escrit instruccions per a afegir més traduccions.
- Escriure una documentació d'ús. Assolit parcialment: al capítol 4 s'inclou una documentació d'ús, però seria necessari penjar-la al lloc web del projecte i traduir-la, com a mínim, a l'anglès.

5.2 ANDRe en el futur

En un futur immediat, l'estratègia a seguir hauria d'incloure els següents punts:

- Manteniment: corregir els errors coneguts fins ara i els que vagin sorgint. Millorar la interfície d'ús. Fomentar l'ús de les eines de seguiment d'errors.
- Ampliació de les funcionalitats: per exemple, en futures versions es podrien implementar les opcions de *desfer* i *refer*.
- Fomentar la col·laboració de la comunitat: promocionar el projecte, millorar-ne el portal web, documentar-lo bé. Cal aconseguir que els usuaris siguin els que informin d'errors i facin suggerències de millora, i que sigui la mateixa comunitat qui corregeixi els errors i millori el programa.

5.3 Nota personal

La realització d'aquest projecte ha fet millorar els meus coneixements i les meves capacitats en alguns aspectes tècnics -notablement, en la programació de fils d'execució en Java-, però sobretot m'ha introduït en la vessant activa de la comunitat de programari lliure.

Tot i que no he aconseguit atraure la comunitat envers ANDRe, degut probablement a que la durada del projecte ha estat molt curta, he après i he dut a la pràctica els primers passos necessaris per a fer-ho. M'he adonat que les tasques d'acostament a la comunitat suposen una gran part de la càrrega de treball total d'un projecte de programari lliure i crec que sovint s'infravalora la seva dificultat i la seva utilitat.

Respecte a l'assoliment d'objectius, considero que se'n pot fer una valoració positiva i que demostra que és realista pensar en una versió estable i funcional d'ANDRe a mitjà termini.

Referències

- [1] PDFBox: <http://pdfbox.apache.org/index.html>
- [2] SourceForge: <http://sourceforge.net/>
- [3] ANDRe: <https://sourceforge.net/projects/projectandre/>
- [4] Text de la llicència GNU GPLv3: <http://www.gnu.org/licenses/gpl.html>
- [5] Popularitat de Java (1): <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [6] Popularitat de Java (2): <http://sourceforge.net/search/?q=>
- [7] PDF-Shuffler: pdfshuffler.sourceforge.net
- [8] PDF Split and Merge: www.pdfsam.org
- [9] PdfMod: live.gnome.org/PdfMod
- [10] PdfJumbler: sourceforge.net/projects/pdfjumbler/
- [11] Apunts de l'assignatura *Desenvolupament de projectes de programari lliure*, de la UOC: http://ocw.uoc.edu/informatica-tecnologia-i-multimedia/desenvolupament-de-projectes-de-programari-lliure/Course_listing
- [12] Sintaxi de *javadoc*: <http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>
- [13] Eclipse: <http://www.eclipse.org/>
- [14] L^AT_EX: <http://www.latex-project.org/>
- [15] iText: www.itextpdf.com
- [16] Pdf-Renderer: java.net/projects/pdf-renderer

- [17] jPod: sourceforge.net/projects/jpodlib
- [18] jPodRenderer: sourceforge.net/projects/jpodrenderer
- [19] JPedal: www.jpedal.org
- [20] Text de la llicència Apache License 2.0: <http://www.apache.org/licenses/LICENSE-2.0.html>
- [21] MiGLayout: www.migcalendar.com/miglayout
- [22] JGoodies jgoodies.com
- [23] Llicències compatibles amb GNU GPL: <http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>
- [24] Article a la Wikipedia anglesa sobre Magic Numbers: http://en.wikipedia.org/wiki/Magic_number_%28programming%29
- [25] Apache Ant: <http://ant.apache.org/>
- [26] Codis d'idioma i país de Java: <http://www.oracle.com/technetwork/java/javase/locales-137662.html>

Apèndix A

Recull de classes, interfícies i enumeracions

Jerarquia de classes

- `main.ANDRe`
- `main.DocumentAdmin`
- `controller.menu_actions.ActionBase`
 - `controller.menu_actions.About`
 - `controller.menu_actions.OpenFile`
 - `controller.menu_actions.Quit`
 - `controller.menu_actions.RemovePage`
 - `controller.menu_actions.SaveFile`
- `controller.menu_actions.ActionsConstants`
- `controller.DragController`
- `controller.DropController`
- `gui.MainWindow`
- `gui.GuiConstants`
- `gui.floatingpanel.FloatingPanelSlider`
- `gui.floatingpanel.FloatingPanelLocateEvent`
- `gui.floatingpanel.FloatingPanel`
 - `gui.floatingpanel.floatingthumbnail.FloatingThumbnail`

- gui.floatingpanel.**FloatingPanelContainer**
 - gui.floatingpanel.floatingthumbnail.**FloatingThumbnailContainer**
- gui.floatingpanel.floatingthumbnail.**FloatingThumbnailSelection**
- document.pdf.**ImmutablePdf**
- document.pdf.**EditablePdf**
- page.pdf.**PdfPage**
- locale.**Messages**

Jerarquia d'interfícies

- document.**ImmutableDocument**
- document.**EditableDocument**
- page.**Page**

Jerarquia d'enumeracions

- gui.floatingpanel.floatingthumbnail.
FloatingThumbnailContainer.FloatingThumbnailContainerType
- document.**DocumentType**